AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1. (Currently Amended) In a networked environment, wherein one or more client computer systems make requests for information from a server computer system, the server computer system providing information in response to the requests from the one or more client computer systems, the server computer system having one or more listen sockets and having a backlog queue for queuing connection requests that the server computer system cannot currently handle, a method of the server computer system reducing denials of service even though the server computer system is experiencing a denial of service attack, the method comprising:

receiving a plurality of connection requests ~~attempting a connection for each connection request received by the server computer system~~ from said one or more client computer systems;

establishing a connection socket for at least one of the plurality of connection requests without placing the connection request in a backlog queue;

for each connection request ~~that~~ for which the server computer system cannot currently ~~handle~~ establish a connection socket, placing the connection request in a the backlog queue without then establishing a connection socket, wherein the backlog queue includes connection requests without regard for whether or not the connection request includes associated request data;

~~monitoring the backlog queue;~~

determining that the backlog queue is being used;

~~resetting one or more connection sockets upon notification that the backlog queue is being used~~

in response to the determination, identifying any connection sockets that have no received request data; and

disconnecting the identified connection sockets.

A1

Claim 2. (Original) The method in accordance with Claim 1, further comprising mapping each connection request to a corresponding listen socket.

Claim 3. (Original) The method in accordance with Claim 2, wherein each listen socket has a corresponding backlog queue.

Claim 4. (Original) The method in accordance with Claim 3, wherein placing the connection request in a backlog queue comprises placing the request in the backlog queue corresponding to the listen socket that the connection request mapped to.

Claim 5. (Currently Amended) The method in accordance with Claim 1, wherein ~~attempting a connection for each connection request received by the server computer system from said one or more client computer systems~~ establishing a connection socket for at least one of the plurality of connection requests is performed using a Winsock module.

Claim 6. (Currently Amended) The method in accordance with Claim 1, wherein ~~attempting a connection~~ establishing a connection socket for at least one of the plurality of connection requests comprises calling a module that accepts connections and waits for associated request data before completing.

Claim 7. (Currently Amended) The method in accordance with Claim 6, wherein the module that accepts connects and waits for associated request data before completing comprises a Winsock()AcceptEx() module.

Claim 8. (Currently Amended) The method in accordance with Claim 1, ~~wherein~~ further comprising monitoring the backlog queue, the determination being made while monitoring the backlog queue ~~comprises calling a module that scans at least the backlog queue for activity~~.

A1

Claim 9. (Current Amended) The method in accordance with Claim [[8]] 12, wherein determining that the backlog queue is being used comprises detecting that the module that scans at least the backlog queue has returned.

Claim 10. (Currently Amended) The method in accordance with Claim [[8]] 12, wherein the module that scans at least the backlog queue for activity comprises a Winsock()sclect() module.

Claim 11. (Original) The method in accordance with Claim 10, wherein determining that the backlog queue is being used comprises detecting that the Winsock()select() module has returned.

Claim 12. (Currently Amended) The method in accordance with Claim [[1]] 8, wherein monitoring the backlog queue comprises calling a module that scans at least the backlog queue for activity ~~resetting one or more connection sockets upon notification that the backlog queue is being used comprises the following:~~

~~identifying any connection sockets that have connections but no received request data; and~~

~~disconnecting the identified connection sockets.~~

Claim 13. (Currently Amended) The method in accordance with Claim [[12]] 1, wherein identifying any connection sockets that have connections but no received request data comprises the following:

calling a module that identifies the state of the connection socket.

Claim 14. (Original) The method in accordance with Claim 13, wherein the module that identifies the state of the connection socket comprises a Winsock()getsockopt() module.

Claim 15. (Currently Amended) The method in accordance with Claim 1, further comprising:

A1

specifying a grace period ~~between~~ spanning the time the backlog queue is determined to be used and the time the identified sockets are disconnected, wherein the disconnection is performed only if the backlog queue still has entries after the grace period ~~one or more connection sockets are reset to allow the server computer system to empty the backlog queue, wherein the resetting of the one or more connection sockets is performed only if the backlog queue still has entries after the grace period~~.

Claim 16. (Cancelled)

Claim 17. (Currently Amended) A computer program product for use in a networked environment, wherein one or more client computer systems make requests for information from a server computer system, the server computer system providing information in response to the requests from the one or more client computer systems, the server computer system having one or more listen sockets and having a backlog queue for queuing connection requests that the server computer system cannot currently handle, a computer program product for implementing a method of the server computer system reducing denials of service even though the server computer system is experiencing a denial of service attack, wherein the computer program product comprises computer-executable instructions which, when executed by a processor, implements the following:

receiving a plurality of connection requests ~~attempting a connection for each connection request received by the server computer system~~ from said one or more client computer systems;

establishing a connection socket for at least one of the plurality of connection requests without placing the connection request in a backlog queue;

for each connection request ~~that~~ for which the server computer system cannot currently ~~handle~~ establish a connection socket, placing the connection request in a backlog queue without then establishing a connection socket, wherein the backlog queue includes connection requests without regard for whether or not the connection request includes associated request data;

~~monitoring the backlog queue;~~

determining that the backlog queue is being used;

~~resetting one or more connection sockets upon notification that the backlog queue is being used~~

in response to the determination, identifying any connection sockets that have no received request data; and

disconnecting the identified connection sockets.

Claim 18. (Original) The computer program product in accordance with Claim 17, further comprising computer-executable instructions for mapping each connection request to a corresponding listen socket, wherein each listen socket has a corresponding backlog queue.

Claim 19. (Original) The computer program product in accordance with Claim 17, wherein the computer-executable instructions for placing the connection request in a backlog queue comprise computer-executable instructions for placing the request in the backlog queue corresponding to the listen socket that the connection request mapped to.

Claim 20. (Currently Amended) The computer program product in accordance with Claim 17, wherein the computer-executable instructions for ~~attempting a connection for each connection request received by the server computer system from said one or more client computer systems~~ establishing a connection socket for at least one of the plurality of connection requests comprises at least portions of a Winsock module.

Claim 21. (Cancelled)

Claim 22. (Currently Amended) The computer program product in accordance with Claim 17, further comprising computer-executable instructions for performing the following:

specifying a grace period ~~between~~ spanning the time the backlog queue is determined to be used and the time the identified sockets are disconnected, wherein the disconnection is performed only if the backlog queue still has entries after the grace period ~~one or more connection sockets are reset to allow the server computer system to empty the backlog queue, wherein the resetting of the one or more connection sockets is performed only if the backlog queue still has entries after the grace period~~.

Claim 23. (Cancelled)

A1

Claim 24. (Currently Amended) In a networked environment, wherein one or more client computer systems make requests for information from a server computer system, the server computer system providing information in response to the requests from the one or more client computer systems, the server computer system having one or more listen sockets, each listen socket having a backlog queue for queuing connection requests that the server computer system cannot currently handle, a method of the server computer system reducing denials of service even though the server computer system is experiencing a denial of service attack, the method comprising:

receiving a plurality of connection requests ~~attempting a connection for each connection request received by the server computer system~~ from said one or more client computer systems;

establishing a connection socket for at least one of the plurality of connection requests using a Winsock()AcccptEx() module without placing the connection request in a backlog queue;

mapping each connection request to a corresponding listen socket;

for each connection request ~~that~~ for which the server computer system cannot currently ~~handle~~ establish a connection socket, placing the connection request in the backlog queue corresponding to the listen socket that the connection request mapped to, wherein the backlog queue includes connection requests without regard for whether or not the connection request includes associated request data;

monitoring the backlog queue using a Winsock()select() module;

determining that the backlog queue is being used by detecting that the Winsock()select() module has returned;

identifying any connection sockets that have connections but no received request data using a Winsock()getsockopt() module; and

disconnecting the identified connection sockets.

25. (Currently Amended) The method in accordance with Claim 24, further comprising:

A1

specifying a grace period ~~between~~ spanning the time the backlog queue is determined to be used and the time the identified connection sockets are disconnected, wherein the disconnection is performed only if the backlog queue still has entries after the grace period.